import cv2

import numpy as np

import math

import webbrowser as wb

import os

print ("Enter the name of websites")

print ("fingers for 2")

finger\_2=input()

print (“fingers for 3")

finger\_3=input()

print ("fingers for 4")

finger\_4=input()

tabs=0

count=0

vid = cv2.VideoCapture(0) **// provides an interface to capture live stream with camera**

while(vid.isOpened()): // cap = vid, ret = r , img = b, crop\_img = Croppedimg, grey=color\_grey , value = kernel

r, b = vid.read() **//ret:- "Ret" will obtain return value from getting the camera frame, either true of false.**

cv2.rectangle(b, (400,400), (100,100), (0,255,0),0) /**/Syntax: cv2.rectangle(image, start\_point, end\_point, color, thickness)**

Croopedimg = b[100:400, 100:400] ` **//image manipulater** **resizing aspect ratio**

color\_grey = cv2.cvtColor(Croppedimg, cv2.COLOR\_BGR2GRAY)

kernel = (35, 35) **//kernel size initialized in place of SigmaX and SigmaY**

**blurred = cv2.GaussianBlur(grey, value, 0)**

\_, thresh\_value = cv2.threshold(blurred, 127, 255,

cv2.THRESH\_BINARY\_INV+cv2.THRESH\_OTSU) // thers1 = thresh\_value

(version, \_, \_) = cv2.\_\_version\_\_.split('.')

if version == '4':

con, hr = cv2.findContours(thresh\_value.copy(),cv2.RETR\_TREE,

cv2.CHAIN\_APPROX\_NONE) // contours = con, hierarchy = hr

max\_area = max(contours, key = lambda x: cv2.contourArea(x)) //cnt = max\_area

m, n, o, p = cv2.boundingRect(max\_area) // x,y,w,h = m,n,o,p

cv2.rectangle(Croppedimg, (m, n), (m+o, n+p), (0, 0, 255), 0)

hl = cv2.convexHull(max\_area) // hull = hl

points = np.zeros(Croppedimg.shape,np.uint8) //drawing = points

cv2.drawContours(points, [max\_area], 0, (0, 255, 0), 0)

cv2.drawContours(points, [hl], 0,(0, 0, 255), 0)

hl = cv2.convexHull(max\_area, returnPoints=False)# return point false to find convexity defects

gaps = cv2.convexityDefects(max\_area, hl) //defects = gaps

count\_defects = 0

cv2.drawContours(thresh\_points, con, -1, (0, 255, 0), 3)# to draw all contours pass -1

for i in range(gaps.shape[0]):

q, r, s, t = gaps[i,0] // s,e,f,d = q,r,s,t

st = tuple(max\_area[q][0]) // start = st , end = en , far = ar

en = tuple(max\_area[r][0])

ar = tuple(max\_area[s][0])

a = math.sqrt((en[0] - st[0])\*\*2 + (en[1] - st[1])\*\*2)

b = math.sqrt((ar[0] - st[0])\*\*2 + (ar[1] - st[1])\*\*2)

c = math.sqrt((en[0] - ar[0])\*\*2 + (en[1] - ar[1])\*\*2)

finger\_angle= math.acos((b\*\*2 + c\*\*2 - a\*\*2)/(2\*b\*c)) \* 57 //angle = finger\_angle

if finger\_angle <= 90:

count\_defects += 1

cv2.circle(Croppedimg, ar, 1, [0,0,255], -1)

cv2.line(Croppedimg,st, en, [0,255,0], 2)

if count==0:

cv2.putText(b,””, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, 3)

if count\_defects == 1 and count!=2 and tabs<=8:

wb.open\_new\_tab('http://www.'+finger\_2+'.com')

tabs=tabs+1

cv2.putText(b,””,finger\_2, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (255,0,0), 3)

count=2

elif count\_defects == 2 and count!=3 and tabs<=8:

wb.open\_new\_tab('http://www.'+finger\_3+'.com')

tabs=tabs+1

cv2.putText(b, “”, finger\_3, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (0,0,255), 3)

count=3

elif count\_defects == 3 and count!=4 and tabs<=8:

wb.open\_new\_tab('http://www.'+finger\_4+'.com')

cv2.putText(b, “”,finger\_4, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (255,165,0), 3)

tabs=tabs+1

count=4

elif count\_defects == 4 and count!=5:

cv2.putText(b, “”, (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 3, 3)

os.system("taskkill /im chrome.exe /f")

tabs=0

count=5

else:

cv2.putText(b,"", (50, 100),\

cv2.FONT\_HERSHEY\_SIMPLEX, 3, 3)

if count==2:

cv2.putText(b, fingers2, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (255,0,0), 3)

elif count==3:

cv2.putText(b, fingers3, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (0,0,255), 3)

elif count==4:

cv2.putText(b, fingers4, (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (255,165,0), 3)

elif count==5:

cv2.putText(b, "", (50, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 3, 3)

# show appropriate images in windows

mg = np.hstack((points, Croppedimg))

#not necessary to show contours and can be skipped

cv2.imshow('Con', mg)

k = cv2.waitKey(10)

if k == 27:

break